



COBOL COMPILER
VALIDATION SUMMARY REPORT.

VALIDATION NUMBER CCVS74-VSR240

PERMIT FULLY LEGISLE PRODUCTION

1 301 77 12 71p.]

Prepared By:

FEDERAL COBOL COMPILER TESTING SERVICE DEPARTMENT OF THE NAVY WASHINGTON, D.C. 20376

408 438



DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

BIBLIOGRAPHIC DATA	1. Report No.	2.		3. Recipien	t's Accession No.		
4. Title and Subtitle	CCVS74-VSR240						
The second secon	D # 00007/ vi	ED2/MAggianed by	Managar	5. Report D	ate		
	Report # CCVS74-VS		Manager	1JUL77			
H6180 MULTICS COE	SOL -	of Testing)		6.			
7. Author(s)				8. Performing	ng Organization Rept.		
Same as organizati	ion - see 9.			No.	ig organization requ		
9. Performing Organization !				10. Project	Task/Work Unit No.		
	piler Testing Servi	ce					
Department of the				11. Contrac	t/Grant No.		
Washington, D. C.							
12. Sponsoring Organization		,			Report & Period		
Automatic Data Pro	ocessing Equipment	Selection Office		Covered			
Department of the	Navy						
Washington, D. C.	20376			14.			
15. Supplementary Notes							
16. Abstracts							
This Validation S	ummary Report (VSR)	for the Honey	well Multic	es	COBOL		
Compiler Version	2.3 ( Multics	Version M	R4.0	) provide	s a consolidate		
	sults obtained from						
the 1974 COBOL St.	andard (X3.23-1974/	FIPS PUB 21-1).	The compi	ier was v	aridated at		
	el of FIPS PUB 21-1						
the discrepancies	found. These incl	ude an overview	of the val	idation w	nich lists all		
categories of dis	crepancies by level	/module within >	(3.23-19/4)	a sectio	n relating		
the categories of	discrepancies to e	ach of the Feder	cal levels	of the la	nguage; and		
a detailed listin	g of discrepancies	together with th	ne tests wh	ich were	failed.		
					1		
	-						
17. Key Words and Documen	t Analysis. 17a. Descriptor	s					
Porgramming Langu	ages						
Standards	4865						
Compilers							
COBOL							
Verifying				ACCESSION for			
	'awwaatmaga			MTIS	White Section		
Proving Program C				900	Butt Section [		
Software Engineer	ing			UMANNOUNCED			
				JUSTIFICATION			
		DD					
17b. Identifiers/Open-Ended	Terms	レレ	C				
		DEGER	nren	BY	COLUMN TO THE COLUMN		
CCVS		Ultrania (	1111		AVAILABILITY CODES		
CVS		(U) m 26 1	977	Dist. AV.	IL. and/or SPECIAL		
		D) JOT 20 1		0 10	10		
		Illinguation	mill	LALA	9-		
		UULULUU	رقاقا با	10	144		
	00/00	D			MI		
17c. COSATI Field/Group	09/02				A.		
18. Availability Statement			19. Security Cl	ass (This	21. No. of Pages		
			Report) UNCLA	SSIFIED	72		
Release unl	imited.		20. Security Cl	ass (This	22. Price		
			Page UNCLA	SSIFIED			
ORM NTIS-35 (REV. 3-72)							

## COBOL COMPILER VALIDATION

1. Validation Number

CCVS74-VSR240

2. Vendor

Honeywell Information Systems

3. Mainframe

H6180 (Multics)

4. Compiler Identification

Honeywell Information Systems Multics COBOL Version 2.3

Operating System Identification 5.

Multics Release MR4.0

Compiler Validation System Version Number CCVS74 2.0

7. Federal Information Processing Standard Publication

21-1

\*PLEASE NOTE. The Federal COBOL Compiler Testing Service may make full and free public disclosure of the Validation Summary Report (VSR) in accordance with the "Freedom of Information Act" (5 U.S.C. #552). The results of this validation are only for the pose of satisfying United States Government requirements, and apply to the Computer System, Operating System release, and compiler vers. identified in the VSR. The COBOL Compiler Validation System is used to determine, insofar as is practical, the degree to which the subject compiler conforms to the Federal COBOL Standard. Thus, the VSR is necessarily discretionary and judgmental. The United States Government does not represent or warrant that the statements, or any one of them, set forth in the VSR are accurate or complete. The VSR is not meant to be used for the purpose of publicizing the findings summarized therein.

For information concerning this compiler you can contact the vendor's designated representative named below:

Dr. Clair Miller HISI, Honeywell Center 7900 Westpark Drive McLean, Virginia 22101



## TABLE OF CONTENTS

SECTION	1.	INTRODUCTION
	1.1	Purpose of the Validation Summary Report
	1.2	Preparation of the VSR
	1.3	Organization of the VSR
	1.4	Abstract Covering Compliance to American
		National Standard Programming Language COBOL
	1.5	Federal Standard COBOL
	1.6	Use of the VSR
	1.7	Sources of Additional Information
	1.8	Requests for Interpretation
	1.9	Modules and Language Elements Excluded from Testing 13
	1.10	Timeliness of the Validation Summary Report
SECTION	2.	DETAILED EVALUATION OF ERRORS
~	2.1	Nucleus Level 1
	2.2	Nucleus Level 2
-	2.3	Table Handling Level 1
	2.4	Table Handling Level 2
	2.5	Sequential 1-0 Level 1
	2.6	Sequential I-O Level 2
	2.7	Relative I-0 Level 1
	2.8	Relative I-0 Level 2
	2.9	Indexed I-O Level 1
	2.10	Indexed I-0 Level 2
	2.11	Sort-Merge Level 1
	2.12	Sort-Merge Level 2
	2.13	
	2.14	
	2.15	Segmentation Level 2 4
	2.16	
	2.17	Library Level 2
	2.18	Debug Level 1
	2.19	
	2.20	Inter-Program Communication Level 1 48
	2.21	Inter-Program Communication Level 2
	2.22	
	2.23	
SECTION	3.	COMPILER STATUS
	3.1	Federal Standard COBOL
	3.2	American National Standard COBOL 5
SECTION	4.	SOFTWARE ENVIRONMENT
SECTION	E	ASCII VALIDATION
SECTION	٥.	ASCII VALIDATION
APPENDI	x A -	VALIDATION SUMMARY WORKING DOCUMENT

#### SECTION 1. INTRODUCTION

#### 1.1 Purpose of the Validation Summary Report

The purpose of the Validation Summary Report (VSR) is to identify individual COBOL language elements whose implementation does not conform to American National Standard Programming Language COBOL, X3.23-1974, and to federal Standard COBOL as adopted from the American National Standard by Federal Information Processing Standard 21-1 (FIPS PUB 21-1).

#### 1.2 Preparation of the VSR

The Validation Summary Report is prepared by analyzing the results of running the COBOL Compiler Validation System (CCVS). The COBOL Compiler Validation System consists of audit routines containing features of Federal Standard COBOL, their related data, and an executive routine (VP-routine) which prepares the audit routines for compilation. Each audit routine is a COBOL program which includes many tests and supporting procedures indicating the result of the tests.

The testing of a compiler in a particular hardware/operating system environment is accomplished by compiling and executing each audit routine. The report produced by each routine tells whether the compiler passed or failed the tests in the routine. If the compiler rejects some language elements by terminating compilation, giving fatal diagnostic messages, or terminating execution abnormally, then the test containing the code the compiler was unable to process is deleted and the audit routine compilation and execution repeated.

The compilation listings and the output reports of the audit routines constitute the raw data from which the members of the Federal COBOL Compiler Testing Service produce a Validation Summary Report.

#### 1.3 Organization of the VSR

The Validation Summary Report is made up of several sections the contents of which are described below.

- a. Section 2 summarizes the results of the compilation and execution of the programs comprising the COBOL Compiler Validation System. Section 2 is subdivided into a subsection representing each level of each module defined in American National Standard Programming Language COBOL, X3.23-1974. Each subsection contains a list of all of the language elements which must be implemented in order to claim support of that level/module. The list of language elements will be annotated to include a description of both syntax and semantic errors detected during the validation.
- b. Section 3 FIPS PUB 21-1 defines four Federal Levels of the COBOL Standard. Section 3.1 of the VSR lists the discrepancies described in Section 2 by the Federal Level in which the problem occurs. Section 3.2 lists discrepancies for the Report Writer Module, which is not a part of Federal Standard COBOL.
  - c. Section 4 contains information which describes the software environ-

ment in which the compiler was tested. This includes the name and version of the operating system; the implementor-names which were used in the Environment Division of the programs comprising the CCVS; the options used with the compiler; and if applicable, information regarding the use of compiler optimization features.

- d. Section 5 contains the results of the ASCII validation. The purpose of these tests is to ascertain whether magnetic tapes written in ASCII code and with ANSI standard labels, and card decks with ASCII code, can be transported between the system being validated and a foreign computer system.
- e. Appendix A is the Validation Summary Working Document, a working paper resulting from the compilation and execution of the CCVS, and from which the VSR is derived.
- 1.4 Abstract Covering Compliance to American National Standard Programming Language COBOL

Definition of an Implementation of American National Standard Programming Language COBOL (excerpts from X3.23-1974, Chapter 1, Section 1.5).

An implementation is defined to meet the requirements of the American National Standard COBOL specification if that implementation includes a fully implemented specified level of each of the functional processing modules and of the Nucleus as defined in this Standard. It follows from this that, in order to meet the requirements of this Standard, an implementation must:

- a. Not require the inclusion of substitute or additional language elements in the source program, in order to accomplish any part of the function of any of the standard language elements.
- b. Accept all standard language elements contained in a given level of a module which is specified as being included in the implementation, except as specifically exempted (as pertaining to specific hardware components for which support is not claimed). See "Elements that Pertain to Specific Hardware Components" below.

These points are of particular pertinence in two areas:

- (1) There are throughout the American National Standard COBOL specification certain language elements whose syntax, or effect, is specified to be, in part, implementor-defined. While the implementor specifies the constraints on that portion of each element's syntax or rules that is indicated in this Standard to be implementor-defined, such constraints may not include any requirement for the inclusion in the source program of substitute or additional language elements.
- (2) When a function is provided outside the source program that accomplishes a function specified by any particular standard COBOL element, then the implementation must not require, except for Environment Division elements, the specification of that external function in place of or in addition to that standard language  $\epsilon$ lement:

The following qualifications apply to the American National Standard COBOL

#### specification:

- a. There are certain language elements which pertain to specific types of hardware components. In order for an implementation to meet the requirements of this standard, the implementor must specify the minimum hardware configuration required for that implementation and the hardware components that it supports. Further, when support is thus claimed for a specific hardware component, all standard language elements that pertain to that component must be implemented if the module in which they appear is included in the implementation. Language elements that pertain to specific hardware components for which support is not claimed, need not be implemented. However, the absence of such elements from an implementation of American National Standard COBOL must be specified.
- b. An implementation of American National Standard COBOL may include the ENIER statement or not, at the option of the implementor.
- c. An implementation that includes, in addition to a specified level of each of the functional processing modules and of the Nucleus, elements or functions that either are not defined in the American National Standard COBOL specification or are defined in a given level of a standard module not otherwise included in the implementation, meets the requirements of this Standard. This is true even though it may imply the extension of the list of reserved words by the implementor, and prevent proper compilation of some programs that meet the requirements of this Standard. The implementor must specify any optional language (language not defined in a specified level but defined elsewhere in the Standard) or extensions (language elements or functions not defined in this Standard) that are included in the implementation.
- d. In general, the American National Standard COBOL specification specifies no upper limit on such things as the number of statements in a program, the number of operands permitted in certain statements, etc. It is recognized that these limits will vary from one implementation of American National Standard COBOL to another and may prevent the proper compilation of some programs that meet the requirements of this standard.

#### IMPLEMENTOR-DEFINED LANGUAGE SPECIFICATIONS

Florent

The language elements in the following lists depend on implementor definitions to complete the specification of the syntax or rules for the elements.

The elements whose syntax is partly implementor-defined are:

Etement	Implementor berined Aspect
SOURCE-COMPUTER paragraph	computer-name
OBJECT-COMPUTER paragraph	computer-name
MEMORY SIZE clause	integer
alphabet-name	<pre>implementor-name; whether imple- mentor-names are provided.</pre>

Implementar-Dafinad Assact

SPECIAL-NAMES paragraph	implementor-name
ASSIGN clause	implementor-name
VALUE OF clause	<pre>implementor-name; whether implementor- names are provided.</pre>
RERUN clause	implementor-name and the form; the implementor provides at least one of seven specified forms.
CALL and CANCEL statements	relationship between operand and the referenced program.
COPY statement	relationship between library-name text-name, and the library.
ENTER statement	language-name
Margin R	The location.
Area B	The number of character positions.
Qualification	The number of qualifiers; at least five must be supported.

The elements whose effect is partly implementor-defined are:

Element	Implementor-Defined Aspect
alphabet-name	The correspondence between native and foreign character sets.
implementor-name switches	Whether setting can change during execution.
USAGE IS COMPUTATIONAL clause	Representation and whether automatic alignment occurs.
USAGE IS INDEX clause	Representation and whether automatic alignment occurs.
SYNCHRONIZED clause	Whether implicit FILLER positions are generated; their effect on the size of group items and redefining items.
ACCEPT statement	Maximum size of one transfer of data in Level 1 Nucleus.
DISPLAY statement	Maximum size of one transfer of data in Level 1 Nucleus.
Numeric test	Representation of valid sign in the absence of the SIGN IS SEPARATE

clause.

Hardware Component

Comparison of nonnumeric items

Collating sequence, where NATIVE or implementor-name collating sequence is implicitly or explicitly specified.

Arithmetic expressions

Element

Number of places carried for intermediate results.

Elements That Pertain to Specific Hardware Components

The standard language elements in the list that follows pertain to specific types of hardware components. These language elements must be implemented in an implementation of American National Standard COBOL when support is claimed, by the implementor, for the specific types of hardware components to which they pertain, and the module in which they are defined is included in that implementation.

ttement	naroware Component
-SET clause	Device capable of supporting the specified code.
OLTIPLE FILE TAPE clause	Reel
CLOSEREEL/UNIT statement	Reel or mass storage
CLOSENO REWIND statement	Reel or mass storage
OPENREVERSED statement	Reel with the capability of making records available in the reversed order; mass-storage with the capability of making records available in the reversed order.
OFENNO REWIND statement	Reel or mass storage
OPENI-O statement (Sequential I-O only)	Mass storage
OPEN EXTEND statement	Reel or mass storage
REWRITE statement (Sequential I-O only)	Mass storage
SENDBEFORE/AFTER ADVANCING statement	Devices capable of vertical positioning; devices capable of action based on mnemonic-names.
USEI-O (Sequential I-O only)	Mass storage
WRITEBEFORE/AFTER	Devices capable of vertical posi-

ADVANCING

tioning; devices capable of action based on mnemonic-name.

#### 1.5 The Federal COBOL Standard

The COBOL compiler validation results enclosed in this document reflect the degree to which the subject COBOL compiler implements the Federal COBOL Standard. The Federal COBOL Standard is essentially the same as the American National Standard Programming Language COBOL, X3.23-1974, with two exceptions:

The Federal COBOL Standard defines 4 levels and the ANSI Standard defines only the minimum COBOL implementation and the full standard. Low and High levels of the Federal COBOL Standard (see 1.5.1) correspond to the above two ANSI levels (minus the Report Writer module). Two additional levels, low-intermediate and high-intermediate have been included in the Federal Standard between the highest and lowest subsets. These additional levels accommodate hardware which cannot support the full standard, but which is capable of implementing more than the minimum standard.

The Federal COBOL Standard states that the Report Writer Module is not mandatory in any Federal level, but that the specifications contained in X3.23-1974 should be used to the extent practical, consistent with requirements.

The Federal COBOL Standard requires that a compiler contain as a minimum the elements specified in at least one of the Federal levels. No restrictions are imposed on the inclusion of selected features from higher levels or even unique vendor extensions. Compatibility amoung various implementations of a given level containing additional features must be controlled by management imposed standards and restrictions.

#### 1.5.1 Federal Standard COBOL Levels

- a. Federal Standard COBOL specifications are the language specifications contained in American National Standard Programming Language COBOL, X3.23-1974. For purposes of the Federal Standard, the modules defined in X3.23-1974 are combined into four levels. Not all computers are large enough to accommodate a COBOL compiler containing the full ANSI Standard. Therefore, the Federal Government requires that all compilers acquired by its agencies contain as a minimum one of the four Federal levels, depending on machine size, configuration and user needs. The knowledge that all computers will support at least one of these four subsets simplifies the task of developing machine-independent COBOL programs.
- b. The four levels of Federal Standard COBOL are identified as: Low, Low-Intermediate, High-Intermediate, and High. Each Federal Standard COBOL level is composed of either the high or low levels of the nucleus and ten of the eleven functional Processing Modules (FPMs) defined in X3.23-1974. The four Federal Standard COBOL levels are reflected in the following table. The numbers in the table refer to the level within the FPM or nucleus as designated in X3.23-1974, and a dash in the table denotes that the corresponding FPM is omitted.

	Low Level	Low Inter- mediate Level	High Inter- mediate Level	High Level	
VUCLEUS	1	1	?	2	
VOCEEOS			-	-	
FPMs					
TABLE HANDLING	1	1	2	2	
SEQUENTIAL I-0	1	1	2	2	
RELATIVE 1-0	-	1	2	2	
INDEXED I-0	-	-		2	
SORT-MERGE	-	-	1	2	
REPORT WRITER	-	-	-	-	
SEGMENTATION	-	1	1	2	
LIBRARY	-	1	1	2	
DEBUG	-	1	2	2	
INTER-PROGRAM					
COMMUNICATION	-	1	2	2	
COMMUNICATION		-	2	2	

#### 1.5.2 Conformance to Federal Standard COBOL

A compiler implemented in conformance to Federal Standard COBOL must meet at least the following requirements.

a. The implementation must include all of the language elements of at least one of the levels of Federal Standard COBOL.

b. The implementation must meet all of the requirements defined in American National Standard COBOL, X3.23-1974, Section I, paragraph 1.5, Definition of An Implementation of American National Standard COBOL which is provided in section 1.4 of this VSR.

c. The implementation must provide a facility for the user to optionally specify a level of Federal Standard COBOL for monitoring his source program at compile time. The monitoring will be an analysis of the syntax used in a source program against the syntax included in the specified level of Federal Standard COBOL. Any syntax used in the source program that does not conform to that allowed by the user selected level of Federal Standard COBOL will be diagnosed. The syntax diagnosed as not conforming to the specified level will be identified to the user through a diagnostic message on the source program listing. The diagnostic message will contain, at least: (1) The identification of the source program line number in which the nonconforming syntax occurs, (2) the identification of the level of Federal Standard COBOL that supports the syntax or that the syntax is nonstandard COBOL.

#### .1.6. Use of the VSR

The Federal COBOL Compiler Testing Service may make full and free public disclosure of the Validation Summary Report (VSR) in accordance with the "Freedom of Information Act" (5 U.S.C. #552). The results of the validation are only for the purpose of satisfying United States Government requirements, and apply only to the computer system, operating system release, and compiler version identified in the VSR.

The COBOL Compiler Validation System is used to determine, insofar as is practical, the degree to which the subject compiler conforms to the COBOL Standard. Thus, the VSR is necessarily discretionary and judgmental. The United States Government does not represent or warrant that the statements, or any one of them, set forth in the VSR are accurate or complete. The VSR is not meant to be used for the purpose of publicizing the findings summarized therein.

#### 1.7 Sources of Additional Information

FIPS PUB 21-1 defines the Federal COBOL Language Standard. This publication is available from the Office of ADP Standards Management, National Bureau of Standards, Washington, D. C., 20234.

The detailed COBOL language specifications are given in the publication "American National Standard Programming Language COBOL, X3.23-1974", available from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

An explanation of the COBOL Compiler Validation System is contained in the CCVS User's Guide. This document explains how to run the compiler validation system. The User's Guide and a magnetic tape containing a copy of the CCVS programs are available from the National Technical Information Service, Springfield, Virginia, 22151. (Ordering information can be obtained from the Federal COBOL Compiler Testing Service.)

#### 1.8. Requests for Interpretation

Questions regarding this VSR or the CCVS in general should be forwarded to the FCCTS. If any problem cannot be adequately resolved through the FCCTS, the request for interpretation will be forwarded to the Federal COBOL Interpretation Committee for final resolution.

A brochure describing the validation process including the procedures for requesting a validation and resolution of questions involving interpretation of the current Federal Standard is available from the Department of the Navy, Federal COBOL Compiler Testing Service, Washington, D.C. 20376.

#### 1.9 Modules and Language Elements Excluded from Testing

During an official validation, certain CCVS tests may not be used, and certain facilities provided by the subject compiler may not be tested.

#### 1.9.1 Federal Standard CDROL Approved Interpretations

The National Bureau of Standards published in the Federal Register Vol. 41 No. 179, September 14, 1976, an approved interpretation of Federal Standard COBOL as pertains to the evaluation of arithmetic expressions in the COMPUTE statements. This interpretation states that "size of the intermediate result field is implementor-defined."

Since the results of evaluating arithmetic expressions are not predictable, all COMPUIE statements and IF statements containing arithmetic expressions have been removed from the COBOL Compiler Validation System.

#### 1.9.2 Report Writer Module

FIPS PUB 21-1 excludes the Report Writer Module from the Federal COBOL Standard. However, the Report Writer Module is still tested during a validation if support for that module is claimed by the compiler vendor.

#### 1.9.3 Communication Module

Although it is part of Federal Standard COBOL as defined by FIPS PUB 21-1, the Communication Module is not currently tested in the course of an official validation for two specific reasons. First, a large volume of requests for interpretation on this module have been submitted to the cognizant ANSI committee (X3J4) for resolution. Secondly, facilities for testing were insufficient to determine the validity of the Communication Module test programs during the development of CCVS74.

#### 1.9.4 Vendor Omissions or Extensions

Language elements are not tested which have been legitimately omitted from the implementation by the implementor (refer to 1.4). Additionally, no implementor extensions to the standard COBOL language are tested in any way.

### 1.10 Timeliness of the Validation Summary Reports

The timeliness of the Validation Summary Report is important. Compilers and their related operating system software are modified several times a year. The Compiler Validation System used to validate compilers is also updated during the life of the system. Therefore to ensure that the latest version of both the vendor's compiler and the Validation System are the latest officially released versions, check with the:

Director Federal COBOL Compiler Testing Service Department of the Navy Washington, D. C. 20376 (202) 697-1247

Please use the Validation Summary Report number of this report when corresponding with the Testing Service.

#### SECTION 2. DETAILED EVALUATION OF ERRORS.

This section summarizes the results of the compilation and execution of the programs comprising the COBOL Compiler Validation System (CCVS). The version of the CCVS used during this validation is shown inside the front cover of the VSR.

Section 2 is made up of a variable number of subsections. The number of subsections is dependent on the Level of Federal COBOL being validated. There will be a subsection for each level of each module which is validated. If the high level of a module is validated then there will be two subsections for that module; one for the low level and one for the high level.

A validation of the low level of Federal Standard COBOL would result in three subsections being present. One for Nucleus level 1, one for Sequential I-O level 1, and one for Table Handling level 1.

Each error or deviation noted in this section makes reference to a program or functional COBOL module contained in Appendix A (Validation Summary Working Document). This reference provides the documented results of an occurrence of errors/deviataions detected during the running of the CCVS using the compiler within the environment identified within this document. The Validation Summary working Document is presented in sequence by functional module, functional module level and program number as defined below.

Each program in the COBOL Compiler Validation System is identified by a 5-character program name. The name associates the routine with the functional processing module and level of American National Standard Programming Language COBOL tested within the program.

The five character name has the general format XXNMM. The first two characters are alphabetic and identify the functional module tested by the program. The permissable values are:

NC - Nucleus

TH - Table Handling

SQ - Sequential I-0

RL - Relative I-0

IX - Indexed I-0

ST - Sort-Merge

RW - Report Writer

SG - Segmentation

LB - Library

DB - Debug

IC - Inter-Program Communication

CM - Communication

The third character of the audit routine name is either a 1 or 2, and identifies the level of the functional module being tested. Each module and level is represented by several programs. The fourth and fifth characters of the program name are sequence numbers for programs which test features in the same level of the same functional processing module.

As an example, the program name NC210 is the tenth program in the series of

100

routines which test the second level of the Nucleus module.

Description of Section 2.

Each error/deviation is noted by number in the left hand margin opposite the language element in question. This number is used in section 3 to categorize errors by Federal level (See 1.5.1). Inserted directly below the language element is a brief description of the error. To the right of the language element is a page reference to X3.23-1974, American National Standard Programming Language COBOL. The reference at the end of the description of the error is to Appendix A which contains the detailed information collected during the validation. The reference is made up of the routine name followed by an A or B (A for compile time or syntax error and B for execution time or semantic error) and a number which makes the error unique in Appendix A.

14. 2. 1

#### Example:

2.1 Nucleus Level 1

Operational symbols: S V P

11-21

2.1.9

- \* The scaling character 'P' is not permitted in a
- \* PICTURE character-string.

(NC101.A.2)

- 2.2 Sequential I-O Level 1
- 2.1.9 represents the ninth error for Nucleus Level 1
- II-21 represents the page in X3.23-1974 where the language element is defined
- \* Boxes the description of the error/deviation

NC101.A.2 represents:

Program name - NC101

Syntax error - A second error - 2

## 2.1 NUCLEUS LEVEL 1

anguage Concepts											7.	-75
Characters used		word.		•		•	٠.	•	•			-76
0, 1,, 9	101	word		•	•	•		•	•		•	10
A, B,, Z												
- (hyphen or m	inus	)										
Characters used			tuat	io	n -						1.	-65
" quotation m										•		
( left parent		S										
) right paren												
. period												
space												
= equal sign												
Characters used	in e	diti	nq.								I.	-58
B space												
O zero												
+ plus												
- minus												
CR credit												
DB debit												
Z zero suppre												
* check prote												
\$ currency si	gn											
, comma												
. period / stroke												
Separators												-75
The separators			100	-		-		•	•	• •	1.	- ()
allowed	, 56	11 60	Lon	ani		o in in a	d,	are	11	υt		I - 1
Character-string				•	• •	•		•	•			-76
COBOL words .					• •	•	• •	•	•			-76
Not more tha						•	• •	•	•	• •	1	. 0
User-defined											1.	-76
data-name												
Must beg	in w	ith	an a	to	habe	ti	c c	har	ac	ter	1	I -1
Must be											1	I - 1
level-numb												
mnemonic-n	ame											
paragraph-	name											
program-na	me											
routine-na	me											
section-na	me											
System-names								•	•		I.	-78
computer-n												
implemento		me										
language-n												7.0
Reserved wor	ds.			•		•		•	•		1.	-79
Key words												
Optional w											,	-00
ZERO ZERO	con	star	its.	•		-		•	•		1.	-80
SPACE												
HIGH-VAL	116											
HIGH-VAL	CE											

QUOTE		
Special-character words		I-80
		1-80
Nonnumeric literals have lengths from 1		1-00
through 120 characters		
Numeric literals have lengths from 1 thre	bugh	
18 diaits		
PICTURE character-strings		I-82
Comment-entries		I-82
Reference Format		I - 105
Sequence number		I - 105
Area A		I-105
Division header		I-106
Section header		1-106
Paragraph header		1-107
Data Division entries		1-107
		1-105
Paragraphs		1-107
Data Division entries		I-107
Continuation of lines		I-106
Only nonnumeric literals may be continued		11-1
Comment lines		I-108
Asterisk (*) comment lines		
Stroke (/) comment line		
Identification Division		I-94
The PROGRAM-ID paragraph		11-3
The AUTHOR paragraph		11-2
The INSTALLATION paragraph		II-5
The DATE-WRITTEN paragraph		11-2
The SECURITY paragraph		11-2
Environment Division		1-95
The Source-computer paragraph		11-5
computer-name		
The OBJECT-COMPUTER paragraph		11-6
computer-name		
MEMORY SIZE clause		
PROGRAM COLLATING SEQUENCE clause		
The CRECIAL NAMES conserved		11-8
The SPECIAL-NAMES paragraph		11-0
implementor-name IS mnemonic-name		
implementor-name IS mnemonic-name series		
ON STATUS		
CEE CT. T.		
OFF STATUS		
alphabet-name clause		
alphabet-name clause CURRENCY SIGN clause		
alphabet-name clause		
alphabet-name clause CURRENCY SIGN clause DECIMAL-POINT clause		
alphabet-name clause CURRENCY SIGN clause DECIMAL-POINT clause Data Division		1-97
alphabet-name clause CURRENCY SIGN clause DECIMAL-POINT clause  Data Division	: :	11-11
alphabet-name clause CURRENCY SIGN clause DECIMAL-POINT clause  Data Division	: :	II-11 II-12
alphabet-name clause CURRENCY SIGN clause DECIMAL-POINT clause  Data Division		11-11
alphabet-name clause CURRENCY SIGN clause DECIMAL-POINT clause  Data Division	· ·	II-11 II-12

### CCVS74-VSR24D

	The Ch		th IC ra	ro TU ct ch	u Reao	eh F rana se	· c · s · i c · i · i · i · i · i · i · i · i ·	la tre sti	au ri er sy io	send similar be	: : : :	v (m lhu	el ma a a s :	x x x x x ay	c	be or or	nt nt	e a v s y	rs bt ir	· · · · · · · · · · · · · · · · · · ·	e v 3 (	vi	at ct	. e	- a	P c i	I Cotte	) r: m:	s .				IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	I. I. I. I.	-1 -1 -1 -1	17131181818181818181
		3				(	C	u	rr	er	n c	y		si	g	n.	)																			
			a			- (	R																													
		1																																		
	Re	p				er (												h	ar	a	C 1	t e	rs	•	•		•	•	•		•		1	Ι.	- 2	11
			a							CI	10	,		> 1	9	55.	,																			
		2																																		
	( )	*		ne				~ .	2	01	ih	c	+	i t	11	t	ic																1	7	_ :	21
		C																																		21
	The	RE	0	EF	I	NE	S	(	cl	a	15	6		( 11	a	y	-	0	t	ь	e	•	6.5	t									1	] .	- 2	7
	The																											•								5 1
	The	51	N	CH	R	Oh.	I	7 5	D	(	: (	а	us	5 6		(1	n a	У	+	e	i	ab	br	61	i	Э,	t e	d	S	, 41	V (	)	1			
	The																									D	-	•	•		•		1	1 .	- 5	5 5
		S				1	14.	4 8	-	- 1	1) (2	y	ı	U E		C1 1	0.0	1	-		d I		O	6			,									
	The					cl	a	1	s e																								1	Ι.	- 3	36
	1	ite	er	al																																
				-																													1		o c	,
,	Cond																							•	•	1	•	•	•		•		-			. 1
		mp																						•	•			•	•		•		-			1
						or																														. 1
			R			ti																														
						01										AI	V																			
						01																														
			(												0		i c		0.	0		3 n	ds										T	τ.	-1	. >
			0	0 11	D	ar		SC	on	(	f		no	00	n	u	ne	r	i		0 (	a e	ra	n	15		(0	0	e r	-	•		1	,		-
																							26										I	1 -	- 4	.2
		0									n																						I	1 .	- 4	3
						or																														
	The																					•		•	•		•	•	•		•		-			1
		- 1																				1	8		ia	i	t c	•	•				1	1.		, ,
	Ove																									·							I	1 .	-5	1
	The	A	C	FF	1	9	t	a t	te	m e	n	t		(0	n	1)	,	0	ne	3													I	1 .	- 5	3
	The	A	D	. 5	t	at	0	Ti e	n	t																							I	1 -	-5	5
		ier								e	a	-		5.6	r	1	9 5																			
		) ·																																		
		U																																		

SIZE ERROR phrase	
The ALTER statement (only one procedure-name)	11-57
The DISPLAY statement (only one transfer of data)	11-59
The DIVIDE statement	11-61
INIO identifier	
BY identifier/literal	
GIVING identifier	
ROUNDED phrase	
SIZE ERROR phrase	
The ENTER statement	11-63
The EXII statement	11-64
The GO TO statement (procedure-name is required)	11-65
DEPENDING ON phrase	
The IF statement (statements must be imperative)	II-66
ELSE phrase	
The INSPECT statement (only single character	
data item)	11-68
TALLYING phrase	
ALL	
그리다 그래프 하시다. 그리고 그림을 하는 내용 그렇게 되는 사람들이 되었다. 그리고 있는 사람들이 되었다.	
LEADING	
CHARACTERS	
REPLACING phrase	
ALL	
LEADING	
FIRST	
CHARACTERS	
TALLYING and REPLACING phrases	
The MOVE statement	11-74
	11 14
10 identifier	
identifier series	
The MULTIPLY statement	11 - 77
BY identifier	
GIVING identifier	
ROUNDED phrase	
SIZE ERROR phrase	
The PERFORM statement	11-78
	11
procedure-name	
THRU phrase	
TIMES phrase	
The STOP statement	11-85
literal	
RUN	
The SUBTRACT statement	11-89
identifier/literal series	
FROM identifier	
GIVING identifier	
ROUNDED phrase	
SIZE ERROR phrase	

## 2.2 NUCLEUS LEVEL 2

ALL	el	e m	er	its	5 (	of	1	NU	c .	1 - 2		are		a 0	ar	t	o f	2	N	UC	1	.2					
Lan	gua	ge	(	01	106	pt	s.						-					•									1-75
C	har	ac					d	fo	r	our	101	tua	it:	ion	-	-		-									1-65
	,				n m a																						
	;						on																				
c	har							to	r	ari	t	m e	t	C	or	er	at	10	ns	•	-	-	-	•	•	•	1-52
								on																			
	*							at	10	n																	
	/					sic																					
	* *							a t																			
-	har								r	210	3 1	or	15	•	•	•	•	٠	•	•	•	•	•	•	•	•	I-66
	=						0																				
	>							ha	n																		
_	<						an																				. 75
- 3	epa																										1-75
-								,																			11-1
(	har																									•	I-76
																										•	1-76
		US						W		S	•	•	•	•	•	•	•	•	-	•	•	•	•	•	•	•	1-76
								na	me																		
			CI (			nan		-							_ (		- 6				t						
								b ua																			II-1
		D																									11-1
		16.6						ds																			I-80
			1					c			3 (1)	S	•	•	•	•	•	•	•	•	•	•	•	•	•	•	1-80
						DES		ER	UE	2																	
								UE	c																		
								IES																			
						TES	-	150																			
								ra																			
			0					S																			1-79
			0					r									,	. N	•	•		•				•	1-19
								on															,				
				0.				;												C	. On	illi d	,				
				1 /	201			co												D	No	Τ.	,	0	NIC	T	
0	ual	i f	i																								1-87
	G O		•			211	•	•	•	•		•		•	•	•	•	•	•	•	•	•		•		•	1 01
Ref	ore	en c	0	f.		n a t																					I-105
	ont	in	111.	a t :	ior	0 0	of.	Li	ne.			on t			+ 1	0.0	•	f	w.c			an		•	•	•	1 10.
								ra														011		725			11-1
		110														•	•	•	•		•	•	•	•	•	•	11 ,
Ide	nt:	fi	r	a t	ior	2 6	iv	is	io	n .																	1-94
	ne											-			•		•	•		•	•	•	•		•	•	11-4
					,	,	-					- 10	•	•	•	-	•	•	-	•	•	•	•	•	•	•	
Env	ir	nn	101	1 t	0.	iv:	Si	on																			
	he									ran	r	ani	1 -														11-8
								cl															•	•		-	
				er																							
	- 1		. 4																								. 07

The REDEFINES clause (may be nested) 11-27 The RENAMES clause (may be nested) 11-29 data-name data-name HRU data-name The VALUE clause		he data evel-num 01 thro	ber																				II-12 II-17
The VALUE clause		he RENAM	ES C																				
Procedure Division	TI	he VALUE literal literal	-1, -1	lit HRL	era	al- ite	2 ra										•					•	11-36
Comparison of nonnumeric operands (operands of unequal size are allowed)	A t	cedure c rithmeti andition Simple Relat Rel	ivis c ex cond cond ion atio	sion oprexpreditional conal	ession	ior sic	s ns io	n		:		:	:			:	:	:	:		:	:	11-39 11-41 11-41
Complex condition		Cond Cond Sign	NOT: und ition	isor equa n-na diti	al s ame ion	s i z	e	are	e a	110	. ₩ €	d)			:	:	:	:	:	•	:	:	11-44
Abbreviated combined relation condition		Complex Logic Negat	c contact of	ndit oper simp	tion rate	ors	nd	ND.	, 0 ion	R,	ar •	nd •	NC.										11-46
of transfers of data)		Abbrev ultiple	res	dic	omb s i	ine n a	d	re th	lat net	ior ic	st	or	id i	t i en	or		:	:	:	:	:		11-47
GIVING identifier series CORRESPONDING phrase The ALTER statement	T	FROM ph he ADD s	ras	e emer	nt.																		
The COMPUTE statement	T	GIVING CORRESI be ALTER	ide POND R st	nti ING atem	fie ph nen	r s ras	er se																11-57
The DISPLAY statement (no restrictions on the number of transfers of data)	T	ne COMPL identi ROUNDEL	JTE fier	stat sei rase	teme rie	ent s						•	•			•					•	•	11-58
INTO identifier series GIVING identifier series REMAINDER phrase		he DISPI of to UPON ph	LAY rans nras	stat fers	tem s o	ent f (	iat	a),		•		•	٠	•	•								
	T	INTO i	dent ide	ific nti phra	er fie ase	sei r s	rie	s ie:	s							•	•	•	•	•	•	•	

### ccvs74-VSR240

11	ie	I	F	S	t a	at	e	(T) 6	en	t		(n	9	S	t e	d	st	a	t e	m	en	ts	)										11-66
TI	ne	1	NS	P	E (	T		S	t a	t	e	m e	n	t	(	mt	Lt	i	- 0	h	a r	ac	te	r	da	t a	i	t e	MS	)			II-68
	S																																
T	he	V	01	I E		s t	а	t	e n	e	n	t						-				-	•	•		•	•	•	•	•	•		11-74
	0 (	OR	RE	. 5	PI	N	D	I	46	,	D	h r	a	S	9																		
	ne																				-	-	•	•	•	. =	•	•	•	-	•	•	11-77
	B	Y	10	te	nt	ti	f	1	9 1		S	e r	,	6	5																		
	6	-	-																														70
T	ne									t	6	m e	n	t	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	II-78
	U																																
	V																																11-86
T	he												t		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	-	•	•	11-00
	D											S																					
		-	N																														
			0																														11-89
i	he																•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	11 07
			M (																														
																i	5																
			R																														11-91
1	he															•	•	•		•	•	•	•	-	•	•	-	•	•	•	•	•	11 7,
			I									,	2 5																				
			N																														
			0																														
	0	1	2	vt	1	LI	- 4		1	01	1	d.	26																				

### CCVS74-VSRZ4D

## 2.3 TABLE HANDLING LEVEL 1

6
9
9
-2
-5
-6
-6
-11

### 2.4 TABLE HANDLING LEVEL 2

All elements of 1 TBL 1,2 are a part of 2 T	1	010	OMON	t c	c. f	7	TRI	1.2	216	a	part	Of	1	181	7 . (
---	---	-----	------	-----	------	---	-----	-----	-----	---	------	----	---	-----	-------

Pata Division	
The OCCURS clause	111-2
integer-1 TO integer-2 DEPENDING ON data-name	
ASCENDING/DESCENDING data-name	
data-name series	
ASCENDING/DESCENDING series	
Procedure Division	
The SEARCH statement	111-7
VARYING phrase	
AT END phrase	
WHEN phrase	
The SEARCH ALL statement	111-7
AT END phrase	
WHEN phrase	

### ccvs74-vsR240

## 2.5 SEQUENTIAL I-0 LEVEL 1

Lan	at the same of																													
	Use	r-d	efi	ne	d	WO	rd	S	•				•	•	•	•				•	•	•	•	•	•	•	•		1	-76
		til			-																									
		rec																												
	1-0	st	atu	S	•	•	•	-	•	•			•	•	•	-		•	-	•	•	•	•	•	•	•	•	•	IV	-7
				0:																										
Env																													ΙV	- 1
	The																								•	•	•	•		-4
	The	SEL						er	T	ry	•	•	•	•	•	•		•	•	•	•	•	•		•	•	•	•	1 0	-4
		ASS		-	-	-				n +						- 1	-													
		ORG	-		4,00				0.00							-														
		ACC																												
		FIL									-	6.	1 1			Ca	u	5.6												
	The										5.	h																	TV	-6
		RERI						a.	d	3 .	ai	111	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•		0
		SAM		-				150	,																					
		SAM																												
		SAM		ME	H	00	, ,	6.5																						
Dat	a n	ivi	sic	0																										
	Fil				n					W.																_			IV	-9
	The	fi	Le	de	SC	ri	p t	ic	n	6	nt	r	v	Ī					-				-							-10
	The																												IV	-9
	The																												IV	-11
		int																												
		int																												
	The	CO	DE-	SE	1	cl	au	SE	3 .																				IV	-12
	The																												IV	-13
		dat																												
		dat				se	ri	0 5	5																					
	The									au	156	2																	I	1-14
		STA																												
		OMI	TTE	D																										
	The	RE	COF	D	00	NT	AI	N	5	cl	a	15	6																I١	<b>-1</b> 8
		int																												
																													IV	-19
		imp																												
		imp															e	ri	es	5										
Pro	ced	ure	0	ivi	s i	or	1																							
	The	CL	OSE	S	ta	te	me	nt	t	(0	n	Ly	ê	3	s i	na	1	6	f	11	e - 1	na	me	m a	ау	9	ppe	ear		
			in	3	CL	05	, E	S	ta	te	me	en	t)																I	1-20
		REE	L																											
		UNI	T																											
	The	OP	EN	st	a t	en	er	it	(	or	11	y	а	S	in	ot	e	1	il	le:	-n	a m	e	may	y	ap	pe.	ar		
			in	an	C	PE	N	51	t a	t e	m e	en	t)																I	1-24
		INP	UT																											
		OUT	PUT	T																										
		1-0																												
	The																												I	1-28
		INT																												
		AT																												
	The	RE	UP '	TE	C	1 :	1 1 6	o m	or	t.					-				_	-	_	_	-				-	_	I	1-31

## ccvs74-vsR240

	FROM identifier The USE statement
2.5.1	*A WRITE BEFORE ADVANCING ZERO statement does not execute *correctly. The system prints the line and then advances to *a new page. (SQ101 B)  BEFORE/AFTER PAGE

#### 2.6 SEQUENTIAL I-O LEVEL 2 All elements of 1 SEQ 1,2 are a part of 2 SEQ 1,2 Language Concepts I-80 IV-3 Environment Division IV-4 IV-4 SELECT clause OPTIONAL phrase RESERVE integer AREA(S) clause The I-O-CONTROL paragraph. . . . . . . . . . . IV-6 SAME RECORD AREA clause SAME RECORD AREA series MULTIPLE FILE TAPE clause Data Division The file description entry. . . . . . . . . . . . . IV-10 IV-11 integer-1 TO integer-2 RECORDS integer-1 TO integer-2 CHARACTERS The LINAGE clause . . . . . . . . . . . . . . . . IV-15 2.6.1 \*Programs containing the LINAGE clause did not execute correctly. (SQ202 B, SQ213 B, SQ214 B, SQ215 B) FOOTING phrase TOP phrase BOTTOM phrase The VALUE OF clause . . . . . . . . . . . . IV-19 implementor-name IS data-name implementor-name IS data-name series Procedure Division The CLOSE statement . . . . . ..... IV-20 NO REWIND, REMOVAL, or LOCK file-name series The OPEN statement. . . . . . . . . . . . . . . IV-24 TNPUT REVERSED 2.6.2 \*OPEN ... REVERSED is not supported by this system. This is not an \*error as OPEN ... REVERSED is a language element that pertains to \*a specific type of hardware component. See section 1.4. (SG206 A) NO REWIND OUTPUT NO REWIND EXTEND file-name series INPUT, OUTPUT, 1-0, and EXTEND series

	The USE statement
. 3	The WRITE statement
	*WRITE statements in programs containing the LINAGE clause did not *execute correctly. (SQ202 B, SQ213 B, SQ214 B, SQ215 B)
	BEFORE/AFTER identifier LINES BEFORE/AFTER mnemonic-name AT END-OF-PAGE imperative-statement
	*An END-OF-PAGE clause with a WRITE statement executed incorrectly.  *The printed line followed the last detail line, but it should have  *been the first line on a new page. (SQ213 B)

### 2.7 RELATIVE I-0 LEVEL 1

Language Concepts	
User-defined words	1-76
file-name	
record-name	
I-0 status	V-2
Environment Division	
The FILE-CONTROL paragraph	V-5
	V-5
SELECT clause	
ASSIGN 10 implementor-name clause	
ORGANIZATION IS RELATIVE clause	
ACCESS MODE clause	
SEQUENTIAL	
RANDOM	
FILE STATUS clause	
	V-7
RERUN clause	
SAME AREA clause	
SAME AREA series	
Data Division	
File Section	V-10
	V-11
	v-10
	V-12
integer CHARACIERS	
integer RECORDS	
	V-13
data-name	
data-name series	
The LABEL RECORDS clause	V-14
SIANDARD	
OMITTED	
	V-15
integer-1 TO integer-2 CHARACTERS	
	V-16
implementor-name IS literal	•
implementor-name IS literal series	
Procedure Division	
	V-17
WITH LOCK	
file-name series	
	V-19
INVALID KEY phrase	•
	v-20
INPUT	
GUTPUT	
I-0	
file-name series	
INPUL, OUTPUL, and I-O series	
The READ statement.	V-23
THE BURN STOLEMENT	A C 3

INTO identifier	
AT END phrase	
INVALID KEY phrase	
The REWRITE statement	V-26
FROM identifier	
INVALID KEY phrase	
The USE statement	V-30
EXCEPTION/ERROR PROCEDURE	
ON file-name	
ON INPUT	
ON OUTPUT	
ON 1-0	
The WRITE statement	V-32
FROM identifier	
INVALID KEY phrase	

# 2.8 RELATIVE I-0 LEVEL 2

All elements of 1 REL 0,2 are a part of 2 REL 0,2	
Environment Division	
The FILE-CONTROL paragraph.	V-5
The file control entry	
SELECT clause	v - )
RESERVE integer AREA(S) clause	
ACCESS MODE IS DYNAMIC clause	
The I-O-CONTROL paragraph	V-7
SAME RECORD AREA	V - 1
SAME RECORD AREA entries	
Data Division	
The file description entry	V-11
The BLOCK CONTAINS clause	V-17
integer-1 TO integer-2 RECORDS	. 12
integer-1 TO integer-2 CHARACTERS	
The VALUE OF clause	V-1/
implementor-name IS data-name	v 10
implementor-name IS data-name entries	
Procedure Division	
The READ statement	V-23
NEXT RECORD	
The START statement	V-28
KEY 1S phrase	
INVALID KEY phrase	
The USE statement	V-30
EXCEPTION/ERROR PROCEDURE	. 50
ON file-name series	

## ccvs74-vsR240

## 2.9 INDEXED I-0 LEVEL 1

Lai	nquage C	once	ept	S																					
	User-de	fin	ed	wor	ds.																			I	-76
	file	-na	me																						
	reco	rd-	nam	0																					
	I-O sta									_	_													VI	-2
			•	•	•	-				-	•	•													
En	vironmen	t D	ivi	sio	n																				
	The FIL					ra	or	ar	n h															VI	-5
	The fil																Ī			_				VI	-5
	SELE						<i>y</i> •		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	
										- m		- 1	~												
	ASSI													5 6											
	ORGA							Χ.	ED	C	ta	us	е												
	ACCE					156																			
	-	EQU		IAL																					
	R	AND	OW																						
	RECO	RD	KEY	cl	aus	9.6																			
	FILE	SI	ATU	SC	tau	ISE	1																		
	The I-0	-00	NTR	OL	par	an	ra	gp	h															VI	-8
	RERU	N C	Lau	se																					
	SAME				USE	λ.																			
	SAME																								
		23.13		0.0.																					
Do	ta Divis	inn																							
Va	File Se																							VI	-11
	The fil														•	•	•	•	•	•	•	•	-		-12
																	•	•	•	•	•	•	•	-	-11
	The rec																	•	•	•	•	•	•		-13
	The BLO							15	6	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0 1	-13
	inte	100				· K	2																		
	inte																								
	The DAT	AR	ECO	RDS	c	lau	15	9		•		•		•		•	•		•		•	•		VI	-14
	data	-na	me																						
	data	-na	me	ser	rie	S																			
	The LAE	EL	REC	ORD	S	cla	a u	se																VI	-15
	STAN	DAR	D																						
	OMIT	TED																							
	The REC	ORD	CO	NIT	INS	5 (	: 1	au	50															VI	-16
	inte																								
	The VAL																							VI	-17
	impl													-	-		-		-						
	impl													. i .	0 0										
	Tan Lo	ene	nec	1 -1	10 111		10		, ,	е,	0 1				2 3										
0 -		n :																							
PI	ocedure																							V.	-18
	The CLO			i te:	11611	·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		
	WITH																								
	file																								- 20
	The DEL							•	•	•	•	•		•		•	•	•	•	•	•	•	•	VI	-20
	INV																								-
	The OPE	N S	tat	e m (	ent																			VI	-21
	INPU	IT																							
	OUTE	TUP																							
	1-0																								
	file	-na	me	se	rie	S																			
	INP						d	I -	0	56	r	ie	S												
	and the same of	-	-		100	-			-	-															

The READ statement	 		_			V1-24
INTO identifier			•	-	•	
AT END phrase						
INVALID KEY phrase						
The REWRITE statement	 					VI-28
FROM identifier			•	•	•	
INVALID KEY phrase						
The USE statement						VI-32
EXCEPTION/ERROR PROCEDURE			•		•	V. J.
ON file-name						
ON INPUT						
ON OUTPUT						
ON I-0						
The WRITE statement						VI-33
FROM identifier				- T	•	55
INVALID KEY phrase						

2.10 1	NDEXED I-0 LEVEL 2
	All elements of 1 INX 0,2 are a part of 2 INX 0,2
2.10.1	Environment Division The FILE-CONTROL paragraph
2.10.1	*Alternate keys are not implemented on this system.  * (IX205 A, IX206 A, IX207 A, IX208 A)
	WITH DUPLICATES phrase The I-O-CONTROL paragraph VI-8 SAME RECORD clause SAME RECORD AREA series
	Data Division The file description entry
	integer-1 TO integer-2 CHARACTERS  The VALUE OF clause
	Procedure Division The READ statement
	The START statement VI-30  KEY IS phrase INVALID KEY phrase
	The USE statement VI-32  EXCEPTION/ERROR PROCEDURE  ON file-name series

# ccvs74-VSR240

# 2.11 SORT-MERGE LEVEL 1

Lar	igua	g e	0	01	CE	0	t s																							
	Use	r-	d€	fi	ine	e d	W C	or	ds.									•	•		•	•	•	•		•			I ~	76
		fi	Le		100	n e																								
Env	iro																													
	The	F	IL	E-	- 0 (	N	TRO	OL	D.	era	9.0	ra	ph					•												
	The	f	il	e	0	on	tr	ot	91	iti	ry		-									-	•	-	•	•	•	VI	I ~	. 2
		SE	LE	CT		cl	au!	s e																						
		AS	SI	GN		TO	i	np.	Les	ner	nt	or	-n	am	е	ct	au	ISE	,											
Da:	ta D	iv	is	ic	n																									
	Fil																		•	•	•			•		•	•	VI		-
	The	S	or	t -	- HI (	er	g e	f	il	9 (	de	s c	ri	pt	1 (	n	er	tr	` y	•	•	•	•	•	•	•	•	VI		
	The	0	AI	A	R	EC	OR	DS	C	lat	15	e	•		•	•	•	•	•		•	•	•	•	•	•		AI		
	The	15	EC	OF	9.0	C	ON	TA	IN	S	cl	au	s e								•	•	•			•	-	VI	I -	- 7
Pr	oced																													
	The			-				at	e m	en'	t		•		•		•	-		•	•	•	•	•	•	-	•	VI	1 -	-12
							s e																							
	The							t e	m e	n t	-	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	VI	1 -	13
		-					S 6																							
							ra																							
	The																													
							a t																		ıtı	ut				
							ur																							
							n			p	ro	gr	an	)	•		•	•	•	•	•	•	•	•	-	•	•	VI	1.	-14
		KE	Y	di	a t	a -	n a	m e																						
							am				e s																			
							S																							
						-	16																							
							EN							IN	16															
							CE	DU	BE	P	hr	as	0																	
				TH																										
							as																							
		01					0.0	ED	UR	E	ph	ra	SE	,																
				TH																										
		6	IV	IN	6	of	ra	5 0																						

# 2.12 SORT-MERGE LEVEL 2

All e	lements	of 1	SRT	0.5	are	a	part	of	5	SRT	0.2
-------	---------	------	-----	-----	-----	---	------	----	---	-----	-----

Environment Division	
The FILE-CONTROL paragraph	VII-2
The file control entry	V11-2
SELECT clause	
The 1-0-CONTROL paragraph	V11-3
SAME RECORD AREA clause	
SAME SORT/SORT-MERGE AREA clause	
SAME series	
Procedure Division	
The MERGE statement	VII-8
KEY data-name	
data-name series	
ASCENDING series	
DESCENDING series	
mixed ASCENDING/DESCENDING	
COLLATING SEQUENCE phrase	
USING phrase	
OUTPUT PROCEDURE phrase	
THRU	
GIVING phrase	
The SORI statement (multiple SORI statements are	
permitted)	VII-14
COLLATING SEQUENCE phrase	

# 2.13 REPORT WRITER LEVEL 1

						1	II C	CLU	1		15	n	ot	1	mp	re	m e	nt	6 :	1 (	n	10	1 6	,	SY	5 1	. 6	m .
																								-				
1	ang																											
	Us	ser	- d	e f	ine	d	WC	r	S																	]	-	76
		fi	Le	- n	ame	2																						
	SI	ес	ia	l	rec	iis	t e	rs	5										-	•						]	-	81
		LI	NE	- C	000	TE	R								-										VI	Ι.	-	1
		PA	GE	- (	our	TE	R	•	•	•	•	•		•	•	•				-					VI	1 1	-	1
1																												
	Re	epo	rt	S	ect	ic	n												-						VI	I	I -	2
	Ti	ne	fi	Le	d€	250	r	pt	i	on	$\epsilon$	nt	ry												VI	1	-	3
	11	ne	re	po	rt	d€	25	cr	ip	t i	on	е	nt	ry						-					VI	I.	I -	4
	T	9 €	re	po	rt	gr	0 (	(p)	d	9.5	cr	ip	ti	on	$\epsilon$	nt	tr	1							VI	I	-	6
	Ti	ne	BL	00	K (	CON	TA	II	S	C	t a	us	e												VI	1	I -	2
	1	he	03	DE	C	Lai	156	2												-					VI	I.	I -	2
	T	he	0.0	LU	NN	NI	JME	BEF	?	cl	au	5 €													VI	1	I -	2
	T	he	CO	NT	ROI	(	: 1	a u :	s e										-						VI	1	I -	2
		C	at	a -	nai	ne																						
		d	lat	a-	nai	пе	S	er	ie	S																		
		F	IN	AL																								
	T	he	da	ta	-n	a m e	9	cl.	au	s e									-		-			-	VI	I	I -	•3
	1	he	G R	0.0	P .	INI	) I	CA	TE	C	La	us	e							•					VI	1	I -	3
	T	he	LA	BE	L	RF (	0.0	RD:	S	c t	au	SE													VI	I	I -	3
	T	he	LI	NE	N	UMI	BE	?	cl	au	s e						•		-	-		•		-	V	I	I -	.3
		i	int	0.0	er																							
			N	EX	T	PA	GE																					
	Ţ						UP	C	La	US	$\epsilon$						•	•	•	•	•	•	•	•	V	1	1 -	. 5
							a e	r																				
																												7
	T									•	•				•		•	•	•	•	•	•	•	•	V	1	1 -	. )
							IN	ES																				
			-																									
							IL																					
																										7	т.	1
	1															•	•	•	•	•	•	•	•	•	V.	1	1.	- 44
	f was																								14		T	,
																			•	•	•	-	•	•	V.	1 1	1 .	-4
	1															•	•	•	•	-	•	•	•	•	٧.	L	1.	-4
									6	5.6	r.	16	S															
																									14	, ,	,	,
														•		-	•	•	•	•	•	•	•	•	V	1 1	1.	-4
		Data Residue Ti Ti Ti Ti Ti Ti	The	repo Specia LINE PAGE  Data Div Report The fi The re The BL The CO The CO The CO The CO The CO The LI The LI The NE The LI The NE The PA The LI The RE The RE The RE The RE The SO The SO The The SO The The SO The The SO	report Special LINE-CI PAGE-C  Data Divis Report S The file The repo The BLOC The CODE The CODE The COLU The CONT data- data- FINAL FINAL The data The GROU The LABE The LINE integ PLUS NEXT The PAGE integ HEADI FIRST LAST LAST FOOTI The RECO The RECO The RECO The REPO TRESOUR THE SOUR THE SOUR THE TYPE	report-na Special rec LINE-COUN PAGE-COUN Data Division Report Sect The file de The report The report The BLOCK O The CODE-SI The COLUMN The CONTROI data-nai data-nai final FINAL FINAL d The data-n The GROUP The LABEL The LINE M integer NEXT PLUS in The NEXT G integer PLUS in NEXT PA The PAGE c integer	report-name Special regis LINE-COUNTE PAGE-COUNTE Data Division Report Section The file describe report de The report de The report gr The BLOCK CON The CODE-SET The COLUMN NUTTHE CONTROL of data-name data-name data-name FINAL FINAL data The data-name The GROUP INT The LABEL REGIST THE VINE NUMB integer PLUS integer PLUS integer PLUS integer NEXT PAGE The PAGE clainteger PLUS integer PLUS integer PLUS integer NEXT PAGE The PAGE clainteger The NEXT PAGE The PAGE clainteger The THE PAGE clainteger The THE PAGE clainteger The THE PAGE clainteger The THE TYPE clainteger The TYPE clai	report-name Special registe LINE-COUNTER PAGE-COUNTER PAGE-COUNTER  Data Division Report Section The file descri The report desc The report grow The BLOCK CONTA The CODE clause The COLUMN NUMB The CONTROL clause The COLUMN NUMB The CONTROL clause The GROUP INDICA The BLOCK CONTA The COLUMN NUMB The CONTROL clause The GROUP INDICA The LINE MUMBER INTEGER PLUS integer NEXT PAGE PLUS integer NEXT PAGE The NEXT GROUP INTEGER PLUS integer NEXT PAGE The NEXT GROUP INTEGER THE PAGE claus THE PAGE	report-name Special registers LINE-COUNTER. PAGE-COUNTER.  Data Division Report Section. The file descript The report descript The report group The BLOCK CONTAIN The CODE clause The COLUMN NUMBER The CONTROL claus data-name data-name sers FINAL FINAL data-nam The data-name claus The GROUP INDICAT The LINE MUMBER integer NEXT PAGE PLUS integer NEXT PAGE PLUS integer The NEXT GROUP conteger PLUS integer NEXT PAGE The PAGE clause integer LINES HEADING FIRST DETAIL LAST DETAIL LAST DETAIL LAST DETAIL FOOTING The PICTURE claus The RECORD CONTA THE SUM CLAUSE THE SUM CLAUSE THE SUM CLAUSE THE SUM CLAUSE	report-name Special registers LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description of the report description of the report group do the BLOCK CONTAINS of the CODE-SET claus of the CODE-SET claus of the CONTROL clause data-name data-name data-name data-name of the GROUP INDICATE of the LINE NUMBER of the LINE MUMBER of the NEXT PAGE of the NEXT GROUP clause of the NEXT GROUP clause of the NEXT GROUP clause of the PAGE clause of the RECORD CONTAIN of the PICTURE clause of the RECORD CONTAIN OF THE RECO	report-name Special registers LINE-COUNTER PAGE-COUNTER PAGE-COUNTER  Data Division Report Section The file description The report descripti The report group des The BLOCK CONTAINS c The CODE clause The CODE clause The CONTROL clause The COLUMN NUMBER cl The CONTROL clause . data-name data-name series FINAL FINAL data-name s The data-name clause The GROUP INDICATE c The LABEL RECORDS cl The LINE NUMBER clause Integer NEXT PAGE PLUS integer The NEXT GROUP clause integer PLUS integer The NEXT GROUP clause integer PLUS integer NEXT PAGE The PAGE clause Integer LINES HEADING FIRST DETAIL LAST DETAIL LAST DETAIL FOOTING The PICTURE clause . The RECORD CONTAINS The REPORT clause . The SOURCE clause . UPON data-name series The SOURCE clause UPON data-name series The SOURCE clause	report-name Special registers LINE-COUNTER PAGE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report group description The BLOCK CONTAINS clause The CODE clause The COLUMN NUMBER clause The CONTROL clause The CONTROL clause The GROUP INDICATE clause The GROUP INDICATE clause The LABEL RECORDS clause The LABEL RECORDS clause The LABEL RECORDS clause The LINE MUMBER clause The NEXT PAGE PLUS integer NEXT PAGE PLUS integer The NEXT GROUP clause THE STATE THE PAGE clause THE RECORD CONTAINS clause THE RECORD CONTAINS clause THE REPORT clause THE REPORT clause THE SUM Clause	report-name Special registers LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report group description e The BLOCK CONTAINS clause The CODE clause The COLUMN NUMBER clause The COLUMN NUMBER clause The CONTROL clause  data-name data-name series FINAL FINAL data-name series The data-name clause The LABEL RECORDS clause The LABEL RECORDS clause The LINE MUMBER clause integer NEXT PAGE PLUS integer The NEXT GROUP clause integer PLUS integer The PAGE clause integer PLUS integer The PAGE clause The PICTURE clause The RECORD CONTAINS clause The RECORD CONTAINS clause The SOURCE clause The SOURCE clause The SOURCE clause The SUM clause	report-name Special registers LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report group description entry The report group description entry The CODE clause The CODE clause The COLUMN NUMBER clause The CONTROL clause The CONTROL clause The GROUP INDICATE clause The GROUP INDICATE clause The LABEL RECORDS clause The LABEL RECORDS clause The LABEL RECORDS clause The LINE MUMBER clause The LABEL RECORDS clause The LINE MUMBER clause The LABEL RECORDS clause The LINE MUMBER clause The NEXT PAGE PLUS integer The NEXT GROUP clause integer PLUS integer The PAGE clause The RECORD CONTAINS clause The RECORD CONTAINS clause The RECORD CONTAINS clause The RECORD CONTAINS clause The REPORT clause The SUMC clause	report-name Special registers LINE-COUNTER  PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report description entry The BLOCK CONTAINS clause The CODE-SET clause The COLUMN NUMBER clause The CONTROL clause The CONTROL clause The GROUP INDICATE clause The GROUP INDICATE clause The LABEL RECORDS clause The LABEL RECORDS clause The LINE MUMBER clause The LOUS integer NEXT PAGE PLUS integer The NEXT GROUP clause The PAGE clause The RECORD CONTAINS clause The SOURCE clause The SUM clause	report-name Special registers LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report group description entry The BLOCK CONTAINS clause The CODE clause The COLUMN NUMBER clause The COLUMN NUMBER clause The COLUMN NUMBER clause The COLUMN NUMBER clause The GROUP INDICATE clause The BROUP INDICATE clause The LABEL RECORDS clause The MEXT GROUP clause The MEXT GROUP clause The MEXT GROUP clause The PAGE clause The PAGE clause The PAGE clause The PICTURE clause The RECORD CONTAINS clause The REPORT clause The REPORT clause The SOURCE clause The SUM clause	report-name  Special registers    LINE-COUNTER    PAGE-COUNTER  Data Division    Report Section    Ine file description entry    The report description entry    The report group description entry    Ine BLOCK CONTAINS clause    The CODE clause    The COLUMN NUMBER clause    The COLUMN NUMBER clause    The CONTROL clause     data-name     data-name series    FINAL    FINAL data-name series    The data-name clause    The BROUP INDICATE clause    The LABEL RECORDS clause    The LINE NUMBER clause    integer    NEXT PAGE    PLUS integer    The NEXT GROUP clause    integer    PLUS integer    The PAGE clause    integer    The PAGE clause    integer    NEXT PAGE    The PAGE clause    integer    The PAGE clause    integer    The NEXT GROUP clause    integer    The PAGE clause    integer    The PAGE clause    integer LINES    HEADING    The PAGE clause    integer LINES    HEADING    The PAGE Clause    The SOURCE clause	report-name Special registers LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report description entry The report group description entry The BLOCK CONTAINS clause The CODE clause The COUF clause The COUNTAINS clause The CONTROL clause data-name data-name series FINAL FINAL data-name series The GROUP INDICATE clause The GROUP INDICATE clause The LABEL RECORDS clause The LABEL RECORDS clause The LINE MUMBER clause integer NEXT PAGE PLUS integer NEXT GROUP clause integer PLUS integer NEXT PAGE The PAGE clause integer LINES MEADING FIRST DETAIL LAST DETAIL LAST DETAIL LOGITING The PICTURE clause The RECORD CONTAINS clause The REPORT clause report-name series The SOURCE clause The SOURCE clause The SOURCE clause The SUM clause	report-name Special registers LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report description entry The report group description entry The BLOCK CONTAINS clause The CODE clause The COLUMN NUMBER clause The COLUMN NUMBER clause The COLUMN NUMBER clause The CONTROL clause data-name data-name series FINAL FINAL data-name series The data-name clause The GROUP INDICATE clause The LABEL RECORDS clause The LINE MUMBER clause integer NEXT PAGE PLUS integer The NEXT GROUP clause integer PLUS integer NEXT PAGE The PAGE clause integer PLUS integer NEXT PAGE The PAGE clause The RETAIL LAST DETAIL LAST DETAIL LAST DETAIL FOOTING The PICTURE clause The RECORD CONTAINS clause The REPORT clause report-name series The SOURCE clause The SUM clause	report-name Special registers LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report aroup description entry The BLOCK CONTAINS clause The CODE-SET clause The CODE-SET clause The CONTROL clause The CONTROL clause data-name data-name series FINAL FINAL data-name series The data-name clause The GROUP INDICATE clause The LINE MUMBER clause The LINE MUMBER clause integer NEXT PAGE PLUS integer The NEXT GROUP clause integer PLUS integer PLUS integer NEXT PAGE The PAGE clause integer NEXT PAGE The PAGE clause The LABL LAST DETAIL LOGIUMC The PICTURE clause The RECORD CONTAINS clause The RECORD CONTAINS clause The REPORT clause report-name series The SOURCE clause The SUM clause	report-name Special registers LINE-COUNTER  Data Division Report Section The file description entry The report description entry The report group description entry The BLOCK CONTAINS clause The CODE clause The COLUMN NUMBER clause The COLUMN NUMBER clause The COLUMN NUMBER clause The CONTROL clause data-name data-name series FINAL FINAL data-name series The data-name clause The GROUP INDICATE clause The LABEL RECORDS clause The LINE NUMBER clause integer NEXT PAGE PLUS integer The NEXT GROUP clause integer PLUS integer NEXT PAGE The PAGE clause integer PLUS integer NEXT PAGE The PAGE clause The RECORD CONTAINS clause The SUMCE clause	report-name Special resisters LINE-COUNTER PAGE-COUNTER  Data Division Report Section The file description entry The report description entry The report group description entry The BLOCK CONTAINS clause The CODE clause The CODE-SEI clause The COLUMN NUMBER clause The CONTROL clause data-name data-name series FINAL FINAL data-name series The data-name clause The SROUP INDICATE clause The LABEL RECORDS clause The LABEL RECORDS clause The LANE MUMBER clause integer NEXT PAGE PLUS integer The NEXT GROUP clause integer PLUS integer The PAGE clause integer PLUS integer The PAGE clause The RECORD CONTAINS clause The SOURCE clause The SOURCE clause The SOURCE clause The SOURCE clause	report-name Special registers LINE-COUNTER	Report - name  Special registers  LINE-COUNTER  PAGE-COUNTER  VIII  Data Division  Report Section  The file description entry  The report description entry  The report group description entry  The BLOCK CONTAINS clause  VIII  The CODE clause  The COLUMN NUMBER clause  VIII  The CONTROL clause  data-name  data-name series  FINAL  FINAL data-name series  The data-name clause  The GROUP INDICATE clause  VIII  The LINE MUMBER clause  VIII  The PAGE  PLUS integer  NEXT PAGE  PLUS integer  PLUS integer  NEXT PAGE  The PAGE clause  The PAGE clause  INTEGER  THE PAGE CLAUSE  THE SOURCE CLAUSE  THE SOURCE CLAUSE  THE SOURCE CLAUSE  THE SUM clause  VIII  THE SUM clause  VIII  THE SUM clause  VIII  THE SUM clause  VIII  THE SUM clause  THE TYPE CLAUSE  VIII	report-name Special redisters			

PAGE HEADING (PH)													
CONTROL HEADING (CH)													
DETAIL (DE)													
CONTROL FOOTING (CF)													
PAGE FOOTING (PF)													
REPORT FOOTING (RF)													
The VALUE IS clause .													11-36
The VALUE OF clause .		•	•	•	•	•	•	•	•				VIII-50
THE VALUE OF CEause .	•	•	•	•	•	•	•	•	•	•	•	•	VIII 30
Procedure Division													
The GENERATE statement													VIII-51
report-name													
data-name													
The INITIATE statement													VIII-53
report-name													
The SUPPRESS statement													VIII-54
report-name													
The TERMINATE statement										_			VIII-55
report-name series		•		•								-	
The USE statement													VIII-56
BEFORE REPORTING	•	•	•	•	•	•	•	•	•	•	•	•	50

# 2.14 SEGMENTATION LEVEL 1

Language Concepts User-defined words	•	•	1-76
Procedure Division Segment-numbers			1x-4
Fixed segment-number range 0 through 49 Non-fixed segment-number range 50 through 99			
All sections with the same segment-number must			

1000

2.15	SEGMENTATION LEVEL 2	
	All elements of 1 SEG 0,2 are a part of 2 SEG 0,2	
	Environment Division	
	The OBJECT-COMPUTER paragraph	
	SEGMENT-LIMIT	1x-5
	Procedure Division	
	Segment-numbers	IX-4
	Sections with the same segment-number need not	
	he obvicedly continuous in the source program	

# 2.16 LIBRARY LEVEL 1

Language Concepts								
User-defined words								1-76
text-name								
All divisions								
The COPY statement								x-2

2.17 L	IBRARY LEVEL 2
	All elements of 1 LIB 0.2 are a part of 2 LIB 0.2
2.17.1	
	*Level 2 of the Library module is not implemented on this system.
	Language Concepts User-defined words
	All divisions The COPY statement

# 2.18 DEBUG LEVEL 1

Language Concepts									
Special registers			 						1-80
DEBUG-ITEM									
Environment Division	1								
The SOURCE-COMPUT	ER para	graph							
WITH DEBUGGING	MODE C	lause.	 •	 •	•	 ٠	•	•	x 1 - 3
Frocedure Division									
USE FOR DEBUGGING	statem	ent						-	XI-4

# ccvs74-VSR240

All elements of 1 DEB 0,2 are a part of 2 DEB 0,2  Procedure Division  USE FOR DEBUGGING statement	*Level 2 of the Debug module is not implemented on this system
Procedure Division	
	All elements of 1 DEB 0,2 are a part of 2 DEB 0,2
USE FOR DEBUGGING statement XI-	Procedure Division
	USE FOR DEBUGGING statement XI-
ALL REFERENCES OF identifier series	file-name series
file-name series	rd-name ceries

# 2.20 INTER-PROGRAM COMMUNICATIONS LEVEL 1

Data Division															
Linkage Section	•	•	•	•	•	•	-	•	-	•		•	•		x I I - 2
Procedure Division															
Procedure Division header. USING phrase	٠	•	•	•	•	•	•	•	•	•	-	•	•	•	x 1 I - 4
The CALL statement															x11-5
literal USING data-name series															
The EXIT PROGRAM statement															×11-8

2.21 INTER-PROGRAM (	COMMUNICATIONS LEVEL 2	
----------------------	------------------------	--

ALL	elemen	ts	of	1	IPC	0.2	are	а	part	o f	2	IPC	0.	2
-----	--------	----	----	---	-----	-----	-----	---	------	-----	---	-----	----	---

Procedure Division									
The CALL statement									X 1 1 - 5
identifier									
ON OVERFLOW phrase									
The CANCEL statement.									X I I - 7

# 2.22 COMMUNICATION LEVEL 1

The Communication Module is no	ot curr	ent	 				part of
an official validation. See S							
Language Concepts							
User-defined words			 				1-76
cd-name							
Data Division							
Communication Section							
The communication description	n entry		 •		•		XIII-3
FOR INPUT clause							
END KEY							
MESSAGE COUNT							
MESSAGE DATE							
MESSAGE TIME SYMBOLIC QUEUE							
SYMBOLIC SOURCE							
SYMBOLIC SUB-QUEUE-n							
STATUS KEY							
TEXT LENGTH							
FOR OUTPUT clause							
DESTINATION COUNT							
DESTINATION TABLE							
INDEXED BY							
ERROR KEY							
SYMBOLIC DESTINATION							
STATUS KEY							
TEXT LENGTH							
Procedure Division							
The ACCEPT MESSAGE COUNT stat			•		-		XIII-12
The DISABLE statement			 •		-		XIII-13
INPUT							
OUTPUT							
KEY identifier/literal The ENABLE statement							XIII-15
INPUT	• • • •	•	 •	• •	•	• •	×111 13
OUTPUT							
KEY identifier/literal							
The RECEIVE statement							XIII-17
MESSAGE							
INTO identifier							
NO DATA phrase							
The SEND statement							XIII-20
FROM identifier-1 WITH							
WITH EMI							
WITH EGI							
BEFORE/AFTER ADVANCING							
identifier-3 LINES							
integer LINES							
mnemonic-name							

PAGE

# 2.23 COMMUNICATION LEVEL 2

* *	The Communnication Module is an official validation. See								part of
				2		0	2		
	All elements of 1 COM 0,2 are a	part	0.1	6	COM	0.	-		
	Communication Section								
	The communication description	ent	ry.						 x 1 1 1 - 3
	FOR INPUT								
	INITIAL								
	Procedure Division								
	The DISABLE statement								 XIII-13
	INPUT								
	TERMINAL								
	The ENABLE statement								 XIII-15
	INPUT								
	TERMINAL								
	The RECEIVE statement								 XIII-17
	SEGMENT								
	The SEND statement							-	 XIII-50
	FROM identifier-1								
	WITH identifier-2								
	WITH ESI								

## SECTION 3. COMPILER STATUS

## 3.1 Federal Standard COBOL

Section 1.5 explains the four levels of Federal Standard COBOL and their relation to American National Standard COBOL. This section lists the discrepancies described in Section 2 by the Federal level in which the problem occurs. All errors listed for a lower level are also errors in any higher level, even though they are listed only in the lower level. The paragraph number from Section 2 is used to reference the errors in each Federal level.

## 3.1.1 Low Level

2.5.1 WRITE...BEFORE ADVANCING ZERO statement.

## 3.1.2 Low-Intermediate Level

2.18.1 Debug module level 1 is not implemented.

## 3.1.3 High-Intermediate

2.6.1 LINAGE clause - execution results not correct.

2.6.3 WRITE statements in programs containing LINAGE clause.

2.6.4 WRITE...AT END-OF-PAGE statement.

2.19.1 Debug module level 2 is not implemented.

## 3.1.4 High Level

2.10.1 Alternate keys are not implemented.

2.17.1 Level 2 of the Library module is not implemented.

### 3.2 American National Standard COBOL

Full American National Standard COBOL consists of the entire set of language elements defined in the ANSI COBOL standard (refer to 1.7). It is also the equivalent of high level Federal Standard COBOL plus the Report Writer module. Therefore, this section lists only those discrepancies found while validating the Report Writer Module.

2.13.1 Report Writer module is not implemented.

SECTION 4. SOFTWARE ENVIRONMENT.

The compiler referenced in this document was validated using the software environment described in this section. When using a modification of the described environment, the compiler may or may not continue to conform to the Standard. It should be noted that during the validation process, an attempt is made to validate as many different options as possible.

The use of compiler options, implementor-names in the Environment Division and any form of optimization which is not described in this report could cause the compiler to produce a program that does not perform according to the specifications of Standard COBOL. Only the environment described in this document has been used with this compiler to satisfy the requirements of FIPS PUB 21-1 and FPMR 101-32.1305.1a. (Any deviations which must be corrected as per the referenced FPMR are described in Sections 2 and 3 of this report.)

1. Options or parameters used on the processor call statement for the compiler: The following options/parameters were used during the validation.

Several different options were used during this validation. The options which were used are:

sc - source
map - map, cross\*reference, source

Some of the routines required special options for execution:

The cs (Cobol\_switch) option was required to set the switch settings for NC103 and NC211.

The tu (continue) option was required to override the SIZE ERROR condition in TH210 and ST202.

The sd (sort directory) option was required for ST116.

The nep (no end page) option was required when printing the output reports for SQ101, SQ201, SQ202, SQ212 through SQ215, to override the default page definition.

2. Environment Division implementor-names.

Printer destined files

REPORT-PRINTER

Tape files

XO1-TAPE CATALOG NAME IS "X101"

Sequential Mass-storage files

X14-VIRTUAL

Random Access files

X21-VIRTUAL

Sort files (SD)

X27-VIRTUAL

Switch names

SWITCH-n

Source Computer names

MULTICS

Object Computer names

MULTICS

3. Optimization. The compiler may or may not have optimization features. If there was an optimization feature available, it was used during the validation process (during a separate execution of the Compiler Validation System) to determine if its use causes the compiler to produce a program which does not give the expected results. If the optimization is invoked through the compiler call statement then it is mentioned in paragraph 1 above. If it is invoked through the introduction of syntax in other than the Data and Procedure Divisions of the source program it is shown below. Optimization which would require modification to the Data and Procedure Divisions is not considered in this report in that it is beyond the scope of the use of standard COBOL and the validation process.

There is no specific optimization feature for this compiler.

4. Compiler.

Multics COBOL Version 2.3

5. Operating system.

Multics Release MR4.0

### SECTION 5. ASCII VALIDATION

#### 5.1 Purpose of ASCII Validation

The ASCII Validation is performed by running a sequence of three CCVS74 programs (SQ118, SQ119, SQ120) using special procedures. The purpose of this special run is to validate that the compiler/operating system being tested is capable of processing ASCII code represented on maganetic tape and punched cards that was produced (in accordance with the appropriate American Vational Standard) by another system. There is also a magnetic tape and a card file created during the validation which will be taken to another system for further processing. The purpose is to determine whether the compiler/operating system being tested can also produce ASCII represented on magnetic tape and punched cards which can be processed by a another computer system.

#### 5.2 Applicable ANSI Standards

The ASCII Validation is based on several American National Standards and presumes their support by the compiler/operating system being validated. These are:

- 1. American National Standard Programming Language COBOL X3.23-1974
  - The CODE-SET clause is used to read and write the ASCII files.
  - The PROGRAM COLLATING SEQUENCE clause is used to process the data in ASCII mode as well as native mode.
  - The SIGN...SEPARATE clause is used for signed data and all data is in the DISPLAY (character) mode.
- American National Standard Code for Information Interchange (ASCII) X3.4-1968. (Note that this describes the code, not the labeling and tape recording formats.)
- American National Standard Hollerith Punched Card Code, X3.26-1970.
- American National Standard Magnetic Tape Labels for Information Interchange, X3.27-1969.
- 5. American National Standard Recorded Magnetic Tape for Information Interchange (800 CPI, NRZI), x3.22-1967.
- American National Standard Recorded Magnetic Tape for Information Interchange (1600 CPI, PR), x3.39-1973.

The language of the 1974 COBOL Standard provides the capability to accept, process, and produce ASCII code. The ASCII Standard describes the code insofar as the bit arrangement and configuration, but does not address recording techniques, record formats or any labeling scheme. The 800 CPI, NZRI magnetic tape recording standard was used to establish the recording density and techniques. (1600 CPI, PE based on X3.39-1973 "Recorded Magnetic Tape for Information Inter-

change" could be used under special arrangements.) The tape labeling scheme used in these tests is based on X3.27-1969 but is also compatible with the revision to that tape label standard. Only the VOL1, HDR1, and EOF1 labels are used. The records are fixed length and unblocked.

#### 5.3 ASCII Validation Process

During the validation, the Validation Manager for the Federal COBOL Compiler Testing Service uses the ASCII-encoded magnetic tape and card files in addition to the normal tape files associated with a validation. For the ASCII portion of the validation the following steps are performed:

- 1. The tape file and card deck (produced on another computer system) are used as input to several programs designed to validate whether the system being validated can accept and process the data as defined by the respective standards. Any changes made during this validation to the source programs reading the data are noted below in 5.4.1.
- 2. A tape file and card file are produced during the validation which should prove to be identical to the files described in 1 above. These two files are then processed on a different computer system to determine the degree to which the system being validated supports the ASCII standard. Any changes made during this validation to the source program producing the data are noted below in 5.4.2.

#### 5.4 Results for This Validation

- The Multics system processed the card deck, the ANSI labeled tape, and the unlabeled tape correctly.
- 2. The Multics system produced an ANSI labeled tape and a card deck which were both verified later as being correct in format and code set. The data records on the tape were proceded by VOL1, HDR1 and HDR2 label records and a tape mark; and were followed by a tape mark, EOF1 and EOF2 label records, and a double tape mark. All labels were correct according to the applicable ANSI Standard for magnetic tape labels.

#### APPENDIX A

#### VALIDATION SUMMARY WORKING DOCUMENT

A-1 This appendix is a working paper produced during the validation and documents the results of the compilation and execution of each of the programs comprising the CCVS. The results contained herein are based on the use of the compiler within the Validation Environment identified in this appendix. This appendix (Validation Summary Working Document) is not part of the official Validation Summary Report (VSR) and is not intended to reflect in any way the compiler's usefulness or degree of conformance to the language specifications.

The reader of this appendix should keep in mind that the same problem area may appear in more than one program, but is considered only as one single discrepancy and as such is reflected only once in the body of the VSR. (The VSR will in turn only reference the first occurrence of the problem in the appendix.)

The reference document for COBOL is FIPS PUB 21-1 (x3.23-1974).

VALIDATION ENVIRONMENT

COMPILER IDENTIFICATION: Multics COBOL Version 2.3

COMPUTER SYSTEM:

H6180 (Multics)

OPERATING SYSTEM:

Multics Release MR4.0

# COMMUNICATION LEVEL 1 and LEVEL 2

The Communication programs were not run.

# DEBUG LEVEL 1 and LEVEL 2

The Debug module is not implemented for this compiler and the Debug programs were not run.

# INTER-PROGRAM COMMUNICATION LEVEL 1 IC101 through IC115, IC151 and IC152

A. Compilation

No errors.

B. Execution

No errors.

# INTER-PROGRAM COMMUNICATION LEVEL 2 IC201 through IC208

A. Compilation

No errors.

B. Execution

## INDEXED I-O LEVEL 1

## 1X101 through IX107

- A. Compilation
  - No errors.
- B. Execution

No errors.

## INDEXED I-O LEVEL 2

## IX201 through IX204

- A. Compilation
  - No errors.
- B. Execution

No errors.

## IX205 through IX208

A. Compilation

Alternate keys are not implemented for this compiler. Statements which referenced alternate keys caused fatal diagnostic messages.

8. Execution

The tests referencing alternate keys had to be deleted. The other tests in these programs executed correctly.

## LIBRARY LEVEL 1

## LB101 through LB107

- A. Compilation
  - No errors.
- B. Execution

No errors.

## LIBRARY LEVEL 2

Level 2 of the Library Module is not implemented for this compiler and the level 2 Library programs were not run.

1 m. m. 1 m. 1

## NUCLEUS LEVEL 1

NC101 through NC120, and NC151 through NC165

- A. Compilation
  - No errors.
- B. Execution

No errors.

## NUCLEUS LEVEL 2

NC201 through NC218

- A. Compilation
  - No errors.
- B. Execution

## RELATIVE I-0 LEVEL 1

# RL101 through RL109 and RL151 through RL153

A. Compilation

No errors.

B. Execution

No errors.

# RELATIVE I-0 LEVEL 2

RL201 through RL205

A. Compilation

No errors.

B. Execution

## REPORT WRITER MODULE

The Report Writer module is not implemented on this system and the Report Writer programs were not run.

## SEGMENTATION LEVEL 1

SG101 through SG106

A. Compilation

No errors.

B. Execution

No errors.

## SEGMENTATION LEVEL 2

SG201 through SG204

A. Compilation

No errors.

B. Execution

## SEQUENTIAL I-0 LEVEL 1

#### 50101

A. Compilation

No errors.

B. Execution

WRT-TEST-31 failed. This test is a WRITE ... BEFORE ADVANCING ZERO statement. The system prints the line and then advances to a new page.

## S@102 through S@121

A. Compilation

No errors.

B. Execution

No errors.

## 59151

A. Compilation

No errors.

B. Execution

WRT-TEST-31 failed. This test is a WRITE ... BEFORE ADVANCING ZERO statement. The system prints the line and then advances to a new page. (See SQ101.)

## S0152 and SQ153

A. Compilation

No errors.

B. Execution

## SEQUENTIAL I-O LEVEL 2

#### 50201

A. Compilation

No errors.

B. Execution

No errors.

## 50202

A. Compilation

No errors.

B. Execution

This program contains the following LINAGE clause:

LINAGE IS 50 LINES
WITH FOOTING AT 45
LINES AT TOP 10
LINES AT ROTTOM 6.

WRT-TEST-07 contains a WRITE ... AFTER ADVANCING PAGE statement. The printed line appeared as the last line on a logical page instead of the first line of a new logical page.

The comment lines for WRT-TEST-08 through WRT-TEST-11 were not printed correctly.

#### SQ203 through SQ205

A. Compilation

No errors.

B. Execution

No errors.

## 50206

A. Compilation

OPEN ... REVERSED is not supported by this system. The \*OPT4 M CCVS control card was used to delete all OPEN ... REVERSED statements from the source program.

2.

B. Execution

No errors.

\$0207 through \$0212

A. Compilation

No errors.

B. Execution

No errors.

59213

A. Compilation

No errors.

B. Execution

This program contains the following LINAGE clause:

LINAGE data-name FOOTING data-name TOP data-name BOTIOM data-name.

There were many errors in the output report. The blank line following the first logical page was omitted for WRI-TEST-01. The first logical page was also incorrect for WRI-TEST-02 and WRI-TEST-03. A blank line was missing preceeding the detail lines and was added following the detail lines.

WRT-TEST-04 is missing two blank lines following the detail lines. WRT-TEST-05 drops the footing line and the blank line which is supposed to follow the detail line.

The line printed from an EOP clause for WRT-TEST-D7 appeared immediately following the last detail line. It should have been the first line on the next page.

Many of the print lines were skewed to the left on the printed output.

50214

A. Compilation

No errors.

B. Execution

This program contains the following LINAGE clause:

LINAGE 40 TOP 2.

WRT-TEST-01, WRT-TEST-02 and WRT-TEST-03 all contained errors. An extra line was printed at the bottom of each logical page instead of at the top of a new page.

## SQ215

A. Compilation

No errors.

B. Execution

The comment line for LIN-TEST-01 was not printed correctly. A single line was printed on two lines.

Sa216 through Sa218

A. Compilation

No errors.

B. Execution

## SORT-MERGE LEVEL 1

## ST101 through ST117

- A. Compilation
- No errors.
- B. Execution
  - No errors.

## SORT-MERGE LEVEL 2

## ST201 through ST215

- A. Compilation
  - No errors.
- B. Execution
  - No errors.

TABLE HANDLING LEVEL 1

TH101 through TH111, and TH151 and TH152

A. Compilation

No errors.

B. Execution

No errors.

TABLE HANDLING LEVEL 2

TH201 through TH220

A. Compilation

No errors.

P. Execution

No errors.

200